PointCom: Semi-Autonomous UGV Control with Intuitive Interface

Mitchell M. Rohde*[a], Victor E. Perlin[a], Karl D. Iagnemma[b], Robert M. Lupa[a], Steven M. Rohde[a], James Overholt[c], Graham Fiorani[d]

[a]Quantum Signal LLC, 3741 Plaza Drive, Ann Arbor, MI, USA 48108; [b]Robotic Mobility Group, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA USA 02139, [c]US Army TARDEC Joint Center for Robotics, AMSRD-TAR-R, 6501 E. 11 Mile Rd MS#263, Warren, MI 48397-5000, [d]US Army TARDEC Intelligent Ground Systems, AMSRD-TAR-R, 6501 E. 11 Mile Rd MS#264, Warren, MI 48397

ABSTRACT

Unmanned ground vehicles (UGVs) will play an important role in the nation's next-generation ground force. Advances in sensing, control, and computing have enabled a new generation of technologies that bridge the gap between manual UGV teleoperation and full autonomy. In this paper, we present current research on a unique command and control system for UGVs named **PointCom** (Point-and-Go Command). PointCom is a semi-autonomous command system for one or multiple UGVs. The system, when complete, will be easy to operate and will enable significant reduction in operator workload by utilizing an intuitive image-based control framework for UGV navigation and allowing a single operator to command multiple UGVs. The project leverages new image processing algorithms for monocular visual servoing and odometry to yield a unique, high-performance fused navigation system. Human Computer Interface (HCI) techniques from the entertainment software industry are being used to develop video-game style interfaces that require little training and build upon the navigation capabilities. By combining an advanced navigation system with an intuitive interface, a semi-autonomous control and navigation system is being created that is robust, user friendly, and less burdensome than many current generation systems.

Keywords: robotics, unmanned, teleoperation, visual odometry, interface, control, mechatronics, human factors

1. INTRODUCTION

Unmanned ground vehicles (UGVs) will play an important role in the nation's next-generation ground forces. Current plans call for unmanned systems to perform a wide variety of roles, including robotic mule applications, unarmed/armed reconnaissance, and EOD/IED (improvised explosive device) inspection and disposal. Deployment of mobile robotic systems (such as the Foster Miller TALON and iRobot PackBot systems) in Bosnia, Ground Zero, Afghanistan and Iraq have served as proofs-of-concept of the effectiveness of these types of systems.

Control of UGVs is accomplished remotely, through a command system that allows an operator(s) to receive sensor data from the UGV (or attendant sensors) and send motion commands to the vehicle. One way to classify these command systems is by the level of supervision required by the human operator, ranging from fully autonomous (i.e. very little or no supervision required) to fully teleoperated (i.e. the operator manually controls every aspect of robot motion).

Fully autonomous control approaches have attracted a significant amount of academic and government research during the past 15 years [1][2][3][4]. These methods are attractive due to their potential to reduce or eliminate operator workload. However, robust real-world implementations of these systems have been elusive. One major problem lies in reliably and accurately gathering and interpreting perceptual information, to distinguish traversable areas from hazardous areas [5][6]. Another problem lies in developing navigation algorithms that can combine situational awareness with complex, non-quantitative factors such as high-level mission goals, to allow a UGV to maneuver in an intelligent and strategic fashion. As a result, many current UGVs have substantial difficulty in navigating through terrain that a human operator would navigate through with ease. Despite continued intense research effort, it is unlikely that a fully

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2009**	2. REPORT TYPE	3. DATES COVERED **00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE **PointCom: Semi-Autonomous UGV Control With Intuitive Interface**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Quantum Signal LLC,3741 Plaza Drive,Ann Arbor,MI,48108**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
See Report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**10**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

autonomous solution to UGV control that exhibits comparable effectiveness as a human-controlled system will be developed within the next several years.

In contrast, fully teleoperated control is considered a mature technology. A significant advantage of full teleoperation is that it exploits a human operator's planning and reasoning skills. However, full teleoperation does not leverage a UGV's ability to perform relatively simple tasks such as path tracking, health monitoring, etc. In addition, teleoperating a multi degree-of-freedom UGV is often challenging and burdensome for an operator due to poor command interface designs that have little regard for human factors issues. As a result, command systems developed for full teleoperation are often difficult to use, complex, and non-intuitive. In fact, instead of resulting in a reduction in workload, some UGV systems require more than one operator to control, partly defeating one purpose of unmanned system development.

A desirable "middle ground" solution to UGV control is semi-autonomy, which leverages the planning, reasoning, and situational awareness skills of a human operator while taking advantage of a UGV's ability to reliably perform low-level control tasks. Ideally, such a system would allow a single operator to easily command multiple UGVs. However, there are significant challenges in developing an effective semi-autonomous UGV command system. Primary areas of difficulty lie in 1) operator interface design, and 2) robust robotic navigation algorithm development.

Challenges in operator interface design stem from the difficultly in translating an operator's intended action into robot motion. This is in large part a function of human factors issues. Most current robot command interfaces seem to have evolved from awkward, complex industrial automation controllers, rather than from more modern and intuitive paradigms such as PDAs and video games. In addition, most interfaces do not allow a single operator to command multiple robots while maintaining adequate situational awareness. Again, this is partly due to the difficulty in developing an operator interface that is simple and intuitive.

Challenges in semi-autonomous UGV navigation derive from classical robotic problems related to detecting and avoiding obstacles, maintaining a record of UGV position (i.e. the localization problem), and accurately navigating along a desired path. All of these problems are complicated by the fact that the environment may be harsh and hazard-rich, and the operator may be unskilled and inexperienced. In addition, it has been demonstrated repeatedly in the robotics community that even the current state-of-the art UGV navigation technology is somewhat non-robust when exposed to real-world scenarios (refer to archival results from the DARPA PerceptOR and LAGR (Learning Applied to Ground Vehicles) programs) [6][7]. Further, the most heavily relied upon sensing methods for navigation - global positioning systems (GPS) – can be intentionally or unintentionally jammed or unavailable in urban or semi-urban environments, rendering many GPS-centric navigation methods useless.

This paper presents an overview of our current research in developing a semi-autonomous control system that addresses these challenges and thus provides robust semi-autonomous control of UGVs with reduced operator workload in the near future.

2. METHODOLOGY

To adequately address the issues discussed above, a hybrid of multiple technologies and approaches are required. The three main activities pursued are:

- Development of a highly-intuitive user interface based on gaming techniques;
- Implementation of intelligent algorithms for monocular obstacle detection and avoidance;
- Fused sensor integration into advanced low-level navigation/control techniques to carry out semi-autonomous tasks.

These activities are described below.

2.1.1 PointCom Interface

The navigation system currently under development is termed PointCom (Point-and-Go Command), an illustration of which is presented in Figure 1. PointCom is a semi-autonomous command system that leverages advanced vision technology and interface design. The system will enable significant reduction in operator workload by:

- Developing an innovative, intuitive image-based control framework for UGV navigation;
- Allowing a single operator to command formations of multiple UGVs.

Human computer interface is an important aspect of any robotic control system, particularly those that must be used in time-critical environments. It is suggested that an "ideal interface" would have the following properties:

- Allows an operator to easily command a UGV to travel to a specific location(s);
- Has few controls yet possesses complex functionality;
- Exhibits low cognitive complexity for operation (i.e. it is "usable by children");
- Employs an intuitive and familiar hardware interface;
- Employs an intuitive and familiar software interface.

Market competition in the video game industry over the past 20 years has led to the evolution of interface designs that meet the above specifications. These interfaces allow complex, flexible character/vehicle control in a manner that is comprehensible by children/teens with little instruction. While a number of UGV manufacturers are recognizing this and thus have adapted inexpensive gaming hardware ("PS2" or "Xbox" style controllers) for use with their vehicle, this represents only a minor leveraging of the technology and expertise. How and why buttons and controls are mapped to functions, along with aspects of the correlated onscreen user interface/display, are key to building an interface that reaches beyond the simple "remote control toy" paradigm. Drawing from this paradigm, PointCom leverages the technology and expertise employed in the design of these gaming interfaces to create substantially more intuitive and flexible interfaces for semi-autonomous UGV control than currently exist.

The approach utilizes a PDA, tablet, or other mobile computing device with a touch screen operator interface. The operator is presented with visual information gathered from a wide field-of-view monocular grayscale camera(s) mounted on the UGV, overlaid with relevant controls and vehicle state information. The command interface GUI and overlays leverage innovative visualization techniques from the entertainment software (e.g. video game) industry. The interface would receive data from the UGV via a low-power, secure communication link at an update rate dependant on UGV velocity and available communications bandwidth.

The operator interface would allow rapid switching between two distinct command modes: "Manual Mode" and "Point and Go." In manual mode, teleoperation of the UGV is essentially just that: the visual scene would be overlaid with buttons to command the UGV to turn left or right, move forward or reverse, and stop (see Figure 1). UGV speed is modulated by tapping a button multiple times. Such a mode is useful to easily command both gross and fine adjustment of UGV position. A single operator can control formations of multiple UGVs in manual mode by choosing a formation geometry (i.e. column, wedge, diamond, etc) then command the motion of a designated "leader" UGV. Other UGVs in the formation will then follow the leader UGV, maintaining formation geometry while autonomously avoiding hazards. This mode would allow for significant reduction in workload while maintaining a high degree of operator supervision. The conceptual design of this control framework again derives from the entertainment software industry, where this type "formation keeping" control is common.

In "Point and Go" mode, an operator designates waypoints or sketches a path on the visual scene by tapping regions on the interface screen, and the UGV navigates to these waypoints using a combination of robotic visual servoing and odometric techniques. This element of the system relies on advanced image processing and feature tracking algorithms under development at Quantum Signal along with robust mobile robot navigation algorithms developed at MIT. Point-and-go mode can also leverage the formation keeping control described above.

Point and Go is the functional focus of the current research, since it will be/is useful in reducing operator workload while relaxing several notoriously difficult challenges of full UGV autonomy. In particular, it leverages the human's ability to perform complex scene interpretation and path planning, and tasks the UGV with relatively simple path planning/following and local hazard avoidance.

The PointCom system architecture is designed to be plug-and-play applicable to a range of systems, from man-portable to moderate size (e.g. R-GATOR sized) to large vehicles. The primary UGV sensor is a wide field-of-view monocular camera (or camera array), which can be grayscale and/or IR for nighttime operation. The use of a small number of simple sensors will lead to a robust, reliable system even in challenging environments. System operation does not rely on GPS (which can be denied and is often unavailable in indoor or urban scenarios), stereo vision (which can be sensitive to lighting and weather conditions), or emissive LADAR technology. Also, it does not require the UGV to solve the localization or autonomous path planning problems.

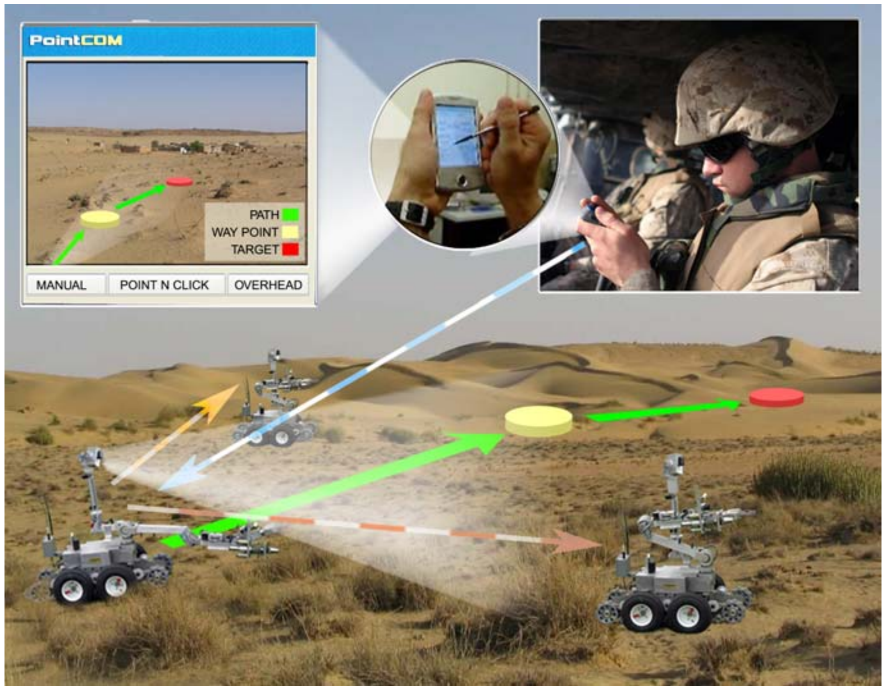

Figure 1: Overview of Proposed PointCom Semi-Autonomous UGV Command System

2.1.2 Image Processing

2.1.3 Implementation of intelligent algorithms for monocular and obstacle avoidance

Most traditional approaches to vision-based outdoor UGV navigation rely upon the use of dense 3-D range data [5][6] derived from stereo. An approach to semi-autonomous UGV navigation based on stereo range data might be divided into four general steps:

1) An operator selects a point (pixel) on the touch-screen interface;

2) The disparity between matching pixels in each stereo image is used to calculate the range to the selected pixel;

3) A path planning algorithm computes a desired path through a local 3-D terrain model from the current UGV position to the selected position;

4) A UGV navigates along the desired path, continuously estimating its pose and position to evaluate its progress.

There are numerous potential pitfalls in this approach. Most notably, the ability to gather dense, accurate range data from stereo is difficult in shadowed regions, direct sunlight, dust, fog, or rain. In addition, range data cannot be collected in visually occluded areas. This often occurs near obstacles (i.e. rocks, trees, etc.) or in regions where the terrain is uneven. (In fact, for a UGV with a 2 m sensor height from the ground, terrain becomes "self-occluded" if it possesses slopes steeper than 65 degrees at 1m range, 12 degrees at 10m range, and only 3 degrees at 30m range.) In general, the reliable collection of range data usually requires a complex and expensive multi-sensor suite [6].

Monocular vision, on the other hand, enjoys the positive aspects of non-emissive, content rich vision sensing while avoiding some of the drawbacks of stereo. In particular, this approach avoids computationally expensive construction of dense 2.5D or 3-D models based on range data. Arguments for the validity of a monocular approach can be found in nature: A dog's narrow eye spacing results in poor stereo vision capabilities, however they can easily move at high speed through complex environments. Instead of performing detailed 3-D geometric analyses, animals clearly perform a significant amount of 2-D scene interpretation. Additionally, humans with strabismus (wandering eye) have limited or no stereo vision yet can perform 3-D tasks (such as driving vehicles) without overwhelming difficulty.

A second potential pitfall to the traditional approach is that it requires a UGV maintain an accurate estimate of its location. This localization problem represents a frontier problem in robotics [8][9][10]. In military scenarios, GPS may not always be available during urban operation or due to enemy countermeasures, and thus the localization problem becomes nearly intractable. Stereo vision-based approaches to GPS-deprived localization (i.e. visual odometry methods) have recently been developed, but these approaches are sensitive to the adverse lighting conditions described above [11].

The PointCom system employs a monocular vision approach that enables UGVs to operate in harsh, real-world conditions without GPS. In this approach, information gained from the motion of the camera is used to simultaneously estimate this motion and the 3D-structure of the environment. Assuming that the terrain surrounding the UGV is relatively flat enables computationally efficient and robust solution of this estimation problem under any type of motion. In the case of rough or sloped terrain, we expect that the inclusion of UGV pitch and roll information (measured by an on-board inclinometer) would improve performance. Such an approach has been successfully employed in rough-terrain environments [12].

2.1.4 Monocular Visual Odometry

Visual odometry is the process of estimating location based on analysis of data from vision sensors. In monocular visual odometry, data from a single camera is used for analysis. While the details of our VO algorithm are beyond the scope of this publication, it is relevant to provide a general overview of the methods.

In a typical robot-eye view, the image can be roughly divided in two distinct regions (Figure 2): near-field (the lower part of the image) and far-field (the upper part). Assuming there are no obstacles, the near-field region represents the ground surface within a few feet in front of the robot, while the far-field region contains the objects that are much further away. As the robot moves, frame-to-frame change (optical flow) of the scene follows two distinct patterns in these two regions (Figure 2, right). The far-field, with the distances to objects much larger than the displacements (physical movements) of the robot, shifts approximately rigidly, almost exclusively due to robot orientation change (translation due to pitch and yaw changes and in-plane rotation due to roll). The near-field, in addition to the same movement as the far-field, exhibits more complex perspective flow patterns due to longitudinal and transverse camera translation. The visual odometry methodology detects and de-couples these flow patterns and thus reconstructs the movement of the robot.

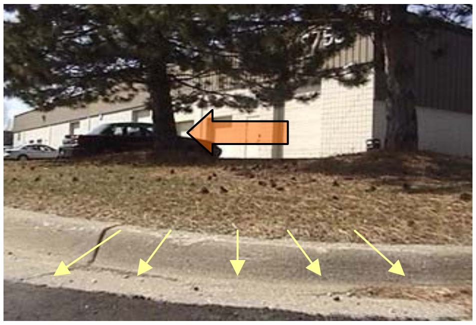

Figure 2: Left: far-field (orange) and near-field (yellow) regions in the camera FOV. Right: typical optical flow patterns in the far-field and near-field areas.

The key component of this system is a novel set of proprietary algorithms developed by QS and termed the “Fast Scene Comparison Framework” (Figure 3). The framework is able to compare any two video frames (not necessarily consecutive) and find the common scene elements independent of their positions (and scale) in their respective frames. Compact representations (~1-3 kB per frame) are computed for each frame, which can be stored and compared quickly (i.e. thousands of comparisons per second). Each new frame is reduced and compared to multiple frames efficiently selected from the recent history. This allows all the scene elements that persist in the FOV be used for robot localization obviating the common problems of standard feature tracking schemes (corrupted frames, lost tracks, mis-tracking, etc). If an object disappears (occlusion, motion blur) in some frames, the system can recover it as soon as it reappears. By

optimizing the selection of frames for comparison, as well as maximizing the number of comparisons attainable in real-time, robust performance on every time step and minimal long-term drift is achieved.

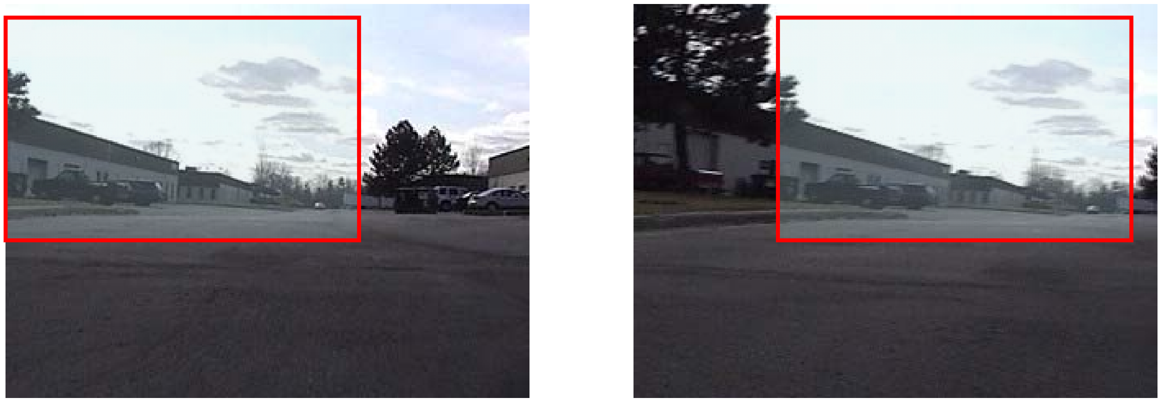

Figure 3: Fast scene comparison framework. Rather than tracking particular objects, the entire common region between 2 frames (separated by a few seconds of robot propagation) is identified.

The QS fast scene comparison framework acts on the far-field regions of FOV and effectively serves as a highly robust angular odometry (i.e. pitch and yaw estimation) module. The angular resolution is determined by video resolution and camera view-angle, and is typically 0.1^{o}-$0.2^{o.}$ for 360x240 video. As with any dead reckoning system, the error accumulates with time or, more precisely, with distance traveled (or with scene changes seen by the robot). The demonstrated rate of uncertainty accumulation is approximately 1^{o}-$5^{o.}$ per minute (depending on surface conditions and robot speeds), similar or lower than published vision-based (including stereo) systems [14],[15] (of course, direct comparisons are not possible as each system was designed for and tested under different set of conditions and constraints) and drastically lower than can be achieved through wheel/track odometry. Furthermore, the described framework has a built-in capability for scene recognition and certainty recovery. If the robot passes the same location twice, it has the ability to recognize the scene as previously observed, and thus reduce orientation uncertainty to that of the previous pass.

Once the robot has precise knowledge of its orientation, near-field optical flow analysis is used to estimate robot displacement and generate its trajectory. As with the angular odometry, at each new frame the current position is referenced not from the immediately preceding frame, but rather from a buffered recent frame, optimally chosen based on the expected displacement between the two frames. This approach enables substantial reduction in the error accumulation rates, as well as provides robustness with respect to blurred or otherwise corrupted frames. Distance error rates typically fall below 3-5% of distance traveled.

In some operating scenarios, far-field information might not be available or simply be insufficient (e.g. indoor scenes with plain walls or outdoor scenes with featureless horizon and sky) for reliable angular odometry. In these cases the scene comparison framework generates low confidence score and the odometry system switches to purely near-field estimation mode (for both rotation and displacement of robot). Under flat-surface assumption (i.e. given that a sufficient fraction of the near-field part of FOV shows the ground surface, as opposed to obstacles, in front of the robot), the whole estimation problem is solved with precision comparable to when far-field information is available. Of course, the uncertainty accumulation rate is increased because the near-field elements of the scene generally remain in the FOV for shorter times than those in the far-field.

2.1.5 Obstacle Detection

A significant challenge in outdoor navigation is the detection and avoidance of obstacles. Though there are many approaches to this based on LIDAR or stereo vision, PointCom exploits feedback from low-cost cameras and intelligent monocular vision algorithms. A monocular camera projects the three-dimensional (3D) world onto a two-dimensional (2D) image by sacrificing range information required to understand the structure of the scene—or, in case of robot navigation, detect obstacles. The only way to recover this information from a single image is to use precise domain

knowledge. One example of such domain knowledge is the “flat surface” assumption mentioned earlier. This assumption, however, has rather limited applicability and, by its own definition, cannot describe obstacles.

There has been recent interest in the research community on more general reconstruction of 3D-scenes or, more narrowly, obstacle detection from single images using machine learning techniques [17],[18] . The ability of such methods, however, to handle scenes that differ substantially from those used in training remains uncertain. More to the point, for a camera mounted on a moving platform, multiple images taken from different locations are readily available, enabling much more promising approach based on structure from motion (SFM).

In its basic form, SFM reconstructs 3D scenes from two 2D images in a way similar to stereo vision. Given the exact relative positions and orientations of the camera when images were gathered, common features in two images are identified and their 3D-positions are found through triangulation. In more sophisticated systems, the exact camera positions and orientations are not known a priori, but are found along with the feature locations, constructing a self-consistent model of the scene. Such approaches often require more than two images at a time, have much higher computational cost, which precludes their real-time implementation, or are subject to degeneracy in 3D feature configurations (e.g. if most features are close to the ground plane, the estimation problem becomes ill-conditioned) [13]). Furthermore, the geometry of the scene can only be determined up to a scale factor, and one still needs some domain information to estimate it. In this work, we approach the problem of obstacle detection via a combination of the techniques described above. Given the imprecision of wheel odometry, especially in describing robot orientation, the relative positions and orientations of the camera when collecting images to be used in SFM are not known in advance, but rather have to be determined from the same images that will be used to reconstruct the 3D-scene (constituting a key step of the visual odometry process). However, to make this problem computationally tractable, the odometry step is de-coupled from the SFM step using domain knowledge, namely a relaxed flat surface assumption. Under this assumption the robot pitch and roll are not required to be zero, but rather small (less than ~1-2°). Also, substantial fraction (i.e. ~25%) of the near-field part of the view must correspond to more or less flat surface in front of the robot, so that consistent optical flow can be computed. Effectively, this means that the robot should not approach obstacles too closely. Calibrated distance from camera to flat surface provides scale information. Once the epipolar geometry estimates are obtained with high confidence (which is also estimated in the process), obstacle detection is performed through the pseudo-stereo SFM analysis.

The main drawback of the monocular system compared to standard stereo vision is the need for precise estimates of the camera shift between the images. Even small errors in these estimates, particularly orientation errors, can result in very noisy obstacle readings. We have employed a two-prong approach to alleviate this problem: a) a comprehensive iterative shift estimation module, and b) accumulation of detection data over time, with signal-to-noise ratio roughly proportional to the number of video frames in which the obstacle is seen. The SFM approach, on the other hand, offers a potential advantage over stereo through the possibility of much longer baselines (typically, just few centimeters for stereo) and hence longer detection range. This advantage could not be realized on the current experimental robot platform, with camera mounted at ~20 inches above ground, because reliable and precise shift estimates could only be achieved at distances ~10 cm (corresponding to detection range ~1-1.5 meters). However, this possibility remains open for larger robotic vehicles with camera mounted higher above ground.

The obstacle detection algorithm, briefly, is a combination of a visual odometry stage (described earlier) and SFM stage. For each current video frame, the optimal reference frame is chosen from a buffer based on the expected relative position considerations. The actual relative positions (translation and rotation) between the current and reference frames are then estimated and passed to the pseudo-stereo obstacle detection module. There the feature points are selected whose shifts along their corresponding epipolar lines could be reliably estimated. By estimating those shifts and triangulating, each feature point is classified as obstacle or not based on its estimated elevation above ground. For obstacle points, the occupancy map scores (see below) are incremented for all cells within the triangulation uncertainty range. The odometry module proceeds with Kalman filtering of the coordinate estimates and fusion of the vision data with the wheel odometry data. The resulting rectified coordinate estimates are used for building the occupancy grid and for robot navigation. Notice, however, that the obstacle detection stage uses only unfiltered data, which is more precise over a single time step. Indeed, pseudo-stereo detection produces meaningful results only with very precise epipolar geometry estimates (with our robot-camera setup, 0.1° orientation error corresponds roughly to 10% distance error, while 0.5° would incur 50% error), achievable only with vision module when reporting high degree of confidence.

2.1.6 Navigation

The "Point and Go" command mode of the PointCom system allows an operator to select a desired waypoint(s) location or path in an image. The UGV then navigates autonomously toward the waypoint using a path tracking algorithm that relies on a combination (fusion) of visual and wheel odometry information for local position estimation. Here it is implicitly assumed that the operator-designated path will be relatively obstacle-free; however, as discussed above, an obstacle detection module based on analysis of monocular imagery is present to detect hazards that the operator might have ignored or that may appear during motion. The navigation algorithm employed in PointCom is summarized as follows:

- At each time step with high confidence score on robot motion estimation, the monocular camera field of view is scanned for obstacles, and detected obstacles are placed in a locally-referenced occupancy map, where each cell's occupancy is incremented or decremented based on the number of cell "hits";
- An obstacle map is generated by thresholding the occupancy scores and dilating to accommodate for finite robot size.
- For the case of designated waypoints (rather than a path), optimal path from the robot's current position to the waypoint(s) is derived via an efficient graph search algorithm (equivalent to D* [16]);

The robot tracks the desired path via a simple pure pursuit-like path tracking algorithm [19]. In the current implementation, the occupancy map covers the area 15x15 meters with cell size 3x3cm. At this resolution, the computational cost of the navigation module is small compared to that of visual odometry, and the system operates smoothly in real time.

3. RESULTS

An initial prototype PointCom system was created and demonstrated in June, 2006, and has been evolving since that time. Different system components and the complete system were tested in several indoor (carpet surface, no far-field features) and outdoor environments (asphalt or grass surface, with or without far-field). The UGV platform used in the experiments is approximately 60 x 60cm in width and length, with the camera mounted 50 cm above ground. The robot was moving at speeds 35-50 cm/sec and turning at 10°/sec. There were two major factors limiting operation to these speeds:

- A relatively low-end camera lens and sensor resulting in image blurring when moving faster (particularly, over bumpy surface) or faster turning.
- The experimental architecture in which the video was transmitted wirelessly from robot to a remote laptop computer where all the vision processing was performed. This resulted in large number of frame drops (often multiple at a time). While the system is relatively robust to occasional dropped frames, moving at faster speeds and multiple consecutive lost frames may result in some performance degradation.

These problems are being addressed in the next version of the system (with better camera, and on-board processing), and it is expected that the equivalent performance at multiples of current speeds will be achieved.

The precision of robot positioning in Point-and-Go mode is largely determined by the precision of the visual odometry module. To measure it accurately would require rather substantial additional hardware or setup to track the movement of the UGV with respect to the outside reference frame [14][15]. In this project a number of ad hoc measurements were performed instead, over a limited number of movement patterns/trajectories for which the ground truth is relatively simple to establish. These include moving along a straight line, circular and square loops, and turning on a spot. On flat surface (e.g. asphalt), the relative distance errors (as percentage of total distance travelled) were in the range ~1-3% for shorter runs (few meters) and somewhat lower ~0.5-1% for longer runs (tens of meters). This is because the errors on each step are independent and tend to average out over longer distances (given there is no systematic error due to miscalibration). On rougher grass surface the errors were approximately double of those above. The heading errors depend on availability of far-field features. In feature-rich outdoor environment the uncertainty accumulation rate is determined by the rate of scene changes and was estimated at ~3°-5° per minute (with actual errors probably lower) for different unconstrained trajectory runs. In such environments, when turning 360° on a square or circular trajectory, the system recognizes the original scene thus avoiding error accumulation. In the opposite case, when the feature-rich far-

field was not available (or simply was not used), the average heading error was ~4° (standard deviation from 360°, measured over multiple runs).

The performance of obstacle detection and avoidance sub-systems is even more difficult to quantify. In general, the more visual features the obstacle has, the faster it is detected and mapped. For feature-poor obstacles the data from multiple video frames has to be accumulated to get high enough occupancy scores. For example, a plain wall with a single horizontal molding line required ~10 frames to be detected. The obstacles without any visual features (e.g. plain wall without molding) would not be detected at all. With the current UGV platform, the maximal distance at which reliable feature triangulation is achievable is ~1.25 m, while the minimal distance is limited by the camera FOV at ~0.8 m (these distances will scale with the size of the platform, in particular with the height of the camera above ground). In our tests, most obstacles (from feature-poor to feature-rich) were detected as expected at distances ~0.8-1.2 meters away.

Obstacle avoidance based on optimal path planning algorithm also performed as expected in our experiments. Generally, the UGV was able to find its way around obstacles to the target pointed by the operator. One limitation, however, has emerged due to the limited detection range of our monocular vision system. In particular, it cannot detect obstacles that are too close to the camera. When the UGV follows the shortest path around an obstacle, it is aware only of the front surface of the obstacle, not of its extents in depth. Accordingly, if the target point is right behind the obstacle, the UGV might start turning into the obstacle's side, which is too close to be detected. A number of approaches to alleviate this problem can be considered: a) constraining path planning algorithm so that only the nearby cells that have been scanned by obstacle detector are allowed; b) modifying the SFM algorithm so it does not require to see ground surface thus reducing the lower bound of the detection range; c) adding some simple additional sensors for short range detection. These possibilities will be investigated in the near future.

4. CONCLUSIONS

In this paper, an overview of PointCom and its components has been presented. A combination of monocular visual odometry and obstacle avoidance, navigation algorithms, and unique interface design has been fused to form a unique semi-autonomous control architecture. Laboratory testing of the system has shown promise, and additional testing in field environments is ongoing.

REFERENCES

[1] Bekey, A., Autonomous Robots: From Biological Inspiration To Implementation And Control, MIT Press, 2005

[2] Brooks, R. and Flynn, A., "Fast, Cheap and Out of Control: A Robot Invasion of the Solar System," Journal of the British Interplanetary Society, 1989

[3] Murphy, R., An Introduction to AI Robotics, MIT Press, 2000

[4] Arkin, R., Behavior Based Robotics, MIT Press, 1998

[5] Bellutta, P., Manduchi, R., Matthies, L., Owens, K., Rankin, A., "Terrain perception for DEMO III," Proc. Intelligent Vehicle Symposium, 2000

[6] Kelly, A., Amidi, O., Bode, M. Happold, M., Herman, H., Pilarski, T., Rander, P., Stentz, A., Vallidis, N., Warner, R., "Toward Reliable Off Road Autonomous Vehicles Operating in Challenging Environments," Proc. of Intl. Symposium on Experimental Robotics, 2004

[7] http://www.darpa.mil/ipto/solicitations/closed/04-25_PIPPrint.htm; also http://www.lagr.net

[8] Thrun, S., Montemerlo, M., Koller, D., Wegbreit, B., Nieto, J., and Nebot, E., "FastSLAM: An Efficient Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association," Journal of Machine Learning, 2004

[9] Thrun, S., and Liu, Y., "Multi-Robot SLAM with Sparse Extended Information Fliers," Proceedings of the 10th International Symposium of Robotics Research, 2003

[10] Bailey, T., Mobile Robot Localization and Mapping in Extensive Outdoor Environments, Ph.D. Thesis, University of Sydney, 2001

[11] Olson, C., Matthies, L., Schoppers, M., and Maimone, M., "Stereo ego-motion improvements for robust rover navigation," Proc. of IEEE Intl. Conference on Robotics and Automation, 2001

[12] See results at: http://www-2.cs.cmu.edu/~deano/research/visualservo/

[13] Hartley, R., Zisserman, A., *Multiple View Geometry*, 2nd.ed., Cambridge University Press, 2003
[14] Nister,D., Naroditsky, O., Bergen, J., "Visual odometry for ground vehicle applications," Journal of Field Robotics, 23(1), 2006
[15] Maimone, M., Cheng, Y., Matthies, L., "Two years of visual odometry on the Mars Exploration Rovers," Journal of Field Robotics, 24(3), 2007
[16] A.Stentz, Optimal and efficient path planning for partially known environments, In Proc. IEEE ICRA, 1994
[17] A. Saxena, S. Chung, A. Ng, 3-D depth reconstruction from a single still image, Int. J. of Computer Vision, 76(1), 2008
[18] Martin, M., Evolving visual sonar: depth from monocular images, Pattern Recognition Letters, 27, 2006
[19] Kelly, A., "A feed-forward control approach to the local navigation problem for autonomous vehicles," Tech. Rep. CMU-RI-TR-94-17, Carnegie Mellon University, 1994.